Presented at the ION GNSS 2008, Network RTK Session, Savannah, Georgia, September 2008

Real-Time, Wide-Area, Precise Kinematic Positioning Using Data from Internet NTRIP Streams

Oscar L. Colombo, *UMBC-GEST/NASA Goddard SFC Code 698*

BIOGRAPHY

Dr. Oscar L. Colombo works on applications of space geodesy for gravity field mapping, orbit determination, and precise positioning by space techniques. He also develops and tests precise long-range positioning techniques using GPS, in collaboration with colleagues in the USA and abroad.

ABSTRACT

A technique for real-time, wide-area differential kinematic positioning at the decimeter level has been tested with data from GPS stations in Europe and the USA, streamed in real time over the Internet by various organizations using the protocol known as NTRIP (Network Transport of RTCM via Internet Protocol).
With the technique presented here, implemented in navigation software developed by the author in cooperation with personnel at NSWC, Dahlgren Division, it is possible, when used in differential navigation mode, to estimate corrections to the GPS broadcast ephemerides in real-time, as part of the navigation solution.
Alternatively, more precise orbits can be used, such as the predicted Ultra-rapid IGS orbits (or IGU orbits), that are updated every few hours, and can be downloaded by anonymous FTP from IGS data repositories such as the CDDIS at NASA's Goddard Space Flight Center.
For this study, 1 Hz data from four stations, three in Spain and one in Portugal, have been used. These stations are separated by distances between 412 and 630 km, and their coordinates are precisely known in the EUREF frame. The data consists of the L1 and L2 carrier phase and pseudo-range observables from the station receivers, and the orbit information in the GPS Navigation Message.
During the test, a station in Madrid, at the center of the network, was positioned kinematically relative to the other three, so the results could be compared to the precise fixed coordinates of the site, used as "truth".
Comparing results obtained with the precise predicted IGS orbits and the uncorrected broadcast orbits shows the errors using the latter to be much larger than with the IGS orbits. On the other hand, the results obtained with the IGS orbits, and those obtained with the broadcast orbits corrected in real time during the navigation solution, are quite similar, with the instantaneous kinematic coordinates differing from their "truth" values at the 10 cm level, after the initial period of convergence of the navigation Kalman filter.

INTRODUCTION

Motivation. Organizations responsible for wide-area SBAS (Satellite Base Augmentation Systems) and CORS (Continuously Operating Reference Stations) are looking for practical ways to add support for precise, decimeter-level, real-time navigation, to their core services in support of reliable meter-level navigation for civil aviation, or cm-level static positioning in post-processing for surveyors. Techniques such as the one presented here would enable wide-area augmentation users to do high-accuracy navigation in real time, in isolated, inaccessible, or remote regions. They could also make this type of service more generally available in developing countries, where the installation and operation of dense networks over large areas may be financially unfeasible. They can make it possible to carry out, for example, precise topographic surveys with airborne lidar or Interferometric SAR (INSAR) over vast expanses of rugged or totally inaccessible terrain, where it would be impractical or impossible to install and operate a dense network of land-based reference receivers.
For precise navigation, it is necessary to correct the data from Global Navigation Satellite Systems (GNSS) such as GPS, GLONASS, or GALILEO, for tropospheric refraction, either with corrections transmitted by the operators of the network, as with the Virtual Reference Stations (VRS) approach, or else estimated from the receiver data as extra unknowns in the navigation solution, as is the case here. And either to use precise ionospheric corrections from a regional or local network

to fix carrier-phase ambiguities [1], or if such are not available, to estimate as real-valued unknowns (float) the corresponding biases in Lc, the ion-free combination of the L1 and L2 carrier phases, e.g. [2], [3].
It is also possible to use the network's receiver data to estimate corrections to the broadcast orbits and achieve with them a level of precision approaching that obtained with precise orbits such as those produced by the International GNSS Service (IGS). To estimate these orbit corrections, one can use a simple orbit perturbation model, based on analytical solutions of Hill's linearized dynamic equations [4], avoiding the need for a CPU-intensive numerical orbit integrator. Based on this simple model one can implement a reduced-dynamics orbit determination procedure to estimate six orbit states and three small unknown stochastic accelerations for each satellite in view [5].
One limitation of this approach, common to all long-baseline procedures, is the time it takes for the navigation Kalman filter solution to converge and produce precise results. Particularly when the residual zenith delay and also the ion-free carrier phase combination biases are estimated ("floating the ambiguities"). Several ways of mitigating this important practical problem have been proposed over the years, e.g. [6].

GNSS data streaming over the Internet, and the NTRIP protocol. The access in real time to GNSS data streamed over the Internet has grown very quickly in the last few years, whether it is free of charge, as provided by governmental, academic, and not-for profit organizations from many countries, or as a paid service by commercial firms. As a result of this, now it is possible for practically anyone with a fast Internet connection, a reasonably fast PC, and the appropriate software, to download and process data from receivers in sites situated all over the world that broadcast more or less continuously in support of real time surveying, navigation, meteorology, etc. This has been helpful to users of precise navigation and to weather services, and has provided a world-wide test-bed for developers of real-time GNSS navigation techniques. In this way, organizations providing freely available Internet data streams are having a positive influence in the improvement and diversification of the uses of GNSS in engineering, science, commerce, and in daily life.
One of the main factors behind this very rapid expansion has been the adoption of the Network Transport of RTCM over the Internet Protocol (NTRIP) [7], [8], [9], [10] which was developed for the Radio Technical Commission for Maritime Services (RTCM) by a group of GNSS vendors. NTRIP Version 1.0 became an RTCM standard in September of 2004. And as of this writing, NTRIP Version 2 has reached the testing stage. This Internet transport protocol facilitates the access to global GNSS data in real time.
NTRIP can be used to send data with RTCM Versions 2.x and Version 3, and also other kinds of data in other formats, such as RINEX.
NTRIP is a subset of the Hypertext Transfer Protocol (HTTP), so it uses the Internet Transfer Control Protocol (TCP). With NTRIP and the related data collection and distribution infrastructure, it is possible to disseminate hundreds of streams simultaneously to thousands of clients distributed all over the world.
NTRIP is used today in a variety of commercial equipment and software applications: RTK with base station and rover receivers, PDA's and mobile phones with integrated GPS receivers, etc. NTRIP data can be streamed over wire, optical fiber, radio link, etc.
As explained in the documentation at the BKG Web site, the NTRIP system consists of four major components:
(1) NTRIP Sources of GNSS data fed into the system. These are, primarily, the GNSS receivers that provide observations or generate RTK correction data.
(2) NTRIP Servers that read data from an NTRIP Source (i.e., a receiver) and send them to an NTRIP Caster.
(3) NTRIP Casters that split incoming data from NTRIP Servers to send it simultaneously to many clients connected to it.
(4) NTRIP Clients are what stationary or mobile users need to access the streamed GNSS data.
Each client chooses a specific NTRIP Source by its ID from an NTRIP Caster. For this, NTRIP includes a Source Table maintained by the NTRIP Caster, describing the content (data rate, RTCM format version, etc) and the Internet ID of any GNSS data stream available.
Information, documentation [11], and NTRIP-related software can be downloaded from the Bundesamt fuer Kartographie und Geodaesie (BKG), in Frankfurt:
http://igs.bkg.bund.de/index_ntrip.htm
Users can get from there free NTRIP-related software. For this work, the author has used the MS Windows version of BKG's client software BNC Version 1.5 to receive NTRIP streams. (There are LINUX and Windows versions.)

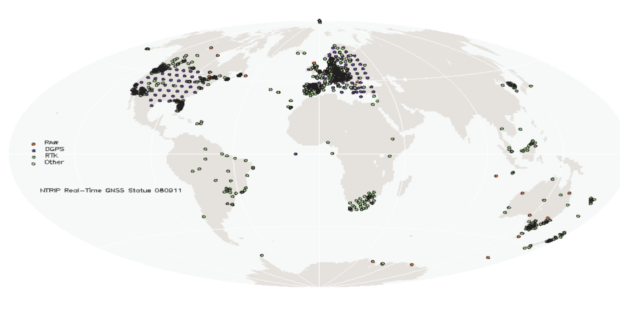

Figure 1. BKG's Map of world-wide sources of NTRIP streams, as of 11 September 2008. Not all streams are publicly available

The map in Figure 1 shows the current locations of many stream sources available through BKG and associated organizations. As of this writing, 1 Hz GPS data is available from many of them. However, there are no guarantees that any particular stream will be available at any given time.
Figure 2 shows one of several dense regional networks whose data are freely accessible on the Internet. This is the *Red GNSS de Castilla y Leon*, in North and Central Spain, operated by the *Instituto de Tecnologia Agraria de Castilla y Leon* (ITACyl).
In the USA, NOAA's NGS is running a test network of three stations in the Chesapeake area [10, *ib.*], not far from the author's home, who has also tested the real-time software with their data.

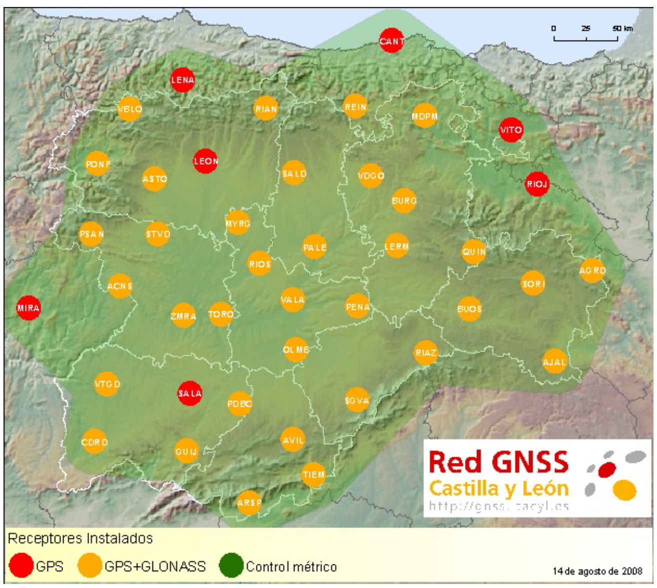

Figure 2. GNSS network of Castilla-Leon. It covers a region of about 450 km by 350 km. (Obtained from http://gnss.itacyl.es/).

In addition to GPS and GLONASS data (L1 and L2 pseudo-range,carrier-phase, Doppler, signal to noise ratios, etc.), some sites also stream precise predicted GNSS orbits from the IGS (ultra-rapid) as well as GNSS navigation data such as broadcast orbits, clock corrections, etc. There are near-future plans to transmit precise clock corrections to allow users to operate also in precise point positioning mode.
From more traditional sources, such as the on-line data and information repositories of the CDDIS at NASA Goddard Space Flight Center, users can, during a real-time navigation session, download files with the latest six-hourly update of the ultra-rapid orbits, and also the "hourly" RINEX navigation file, which is updated every hour or so with the most recent GPS Navigation Message data received in near-real time from GNSS sites around the world.

Data Latency. The delay in transmission of data streamed from a GNSS receiver to a user receiving the stream (which follows a route over the Internet from the receiver to an NTRIP caster server, and from there to the user) is usually well below 10 seconds. Data from a fixed land receiver is highly predictable for up to 20 or 30 seconds. So delayed observations can be predicted by the user's navigation software, with little loss of precision in the results, by fitting a parabola to the last three observations received, computing the value of the parabola at the epoch of the present roving receiver measurements, and using this value in lieu of the actual data. Most of the time the data latency is well below 10 seconds [8, *ib.*], [9, *ib.*], so it can be taken care of in this way.

THE PRECISE POSITIONING TECHNIQUE

Software. The GPS real-time navigation software used in this study has been developed by the author for a project of the US Navy [5, *ib.*], and it is derived from the author's own post-processing software "IT" ("Interferometric Translocation"), with which it shares a number of features, such as its use in either differential or in point-positioning modes. Both the real-time and the post-processing versions of "IT" have been used in the present work.
The navigation technique implemented is based on ideas developed over the last fifty years, first for the geodetic positioning of terrestrial tracking sites and for determining the orbits of artificial satellites, and later adapted by the author and others [2, *ib.*], [3, *ib.*], [12], [13], [14], [15], for kinematic solutions.
The unknowns, or error states estimated with the navigation Kalman filter include, besides the position of the roving receiver and of some of the fixed stations (in differential solutions), also a zenith delay per receiver, the biases in the ion-free linear combination of L1 and L2 carrier phases due to the ambiguities, satellite clock errors (in some solutions), and the orbit parameters of each GPS satellite: three initial position and three initial velocity components, and three random, piece-wise constant residual accelerations, using the reduced-dynamics approach [16].
The orbit part of the solution is explained in this section. Its main characteristics are: the use of analytical orbit perturbations instead of a numerical orbit integrator, greatly reducing the computing overhead, and the use of a pseudo-epoch state formulation, which has a unit transition sub-matrix, in line with the other error states, which have either "white noise", constant, or random-walk dynamics, all chosen to save computing time,
The real-time program implements only a Kalman filter step, which in the off-line version is followed by a smoother step. In both versions, the navigation Kalman filter is updated once every minute if the receiver data rate

is at least one epoch every five seconds, and every two minutes, if slower. Instantaneous and compressed data are used in the updates, to assimilate all the data [12, *ib.*]

In real time, at epochs between filter updates, the values of the filter estimates of the Lc biases, zenith delays, and (when included in the solution) of the orbit errors, are extrapolated to the epoch of the present data with parabolas fitted to their values from the three previous filter updates. The data are corrected with these extrapolated values, and are used to find the position of the rover. This is done solving for corrections to the nominal x, y, z coordinates (previously estimated with the pseudo-range data) as the only unknowns in a small linear least squares solution. The points of the vehicle trajectory are determined at each epoch, the whole process repeating with, and being paced by, the successive arrival of new data. (The corresponding operation in the off-line version involves interpolating between the two consecutive updates bracketing the epoch of the data; this is done after the filter and the smoother steps, to obtain the full-rate kinematic solution in a final step.)

The Adjustment of the Broadcast Orbits:

Main Navigation Kalman filter updates. Updating a Kalman filter involves three steps [17]:

(I) Deterministic update: the full state vector is multiplied by its transition matrix, and the covariance matrix of this vector is pre- and post-multiplied by that matrix. The dimensions of the state vector and the covariance matrix, in precise solutions, can be in the hundreds [2, *ib.*], and the vector and matrix products just described could mean a heavy computing load that slows down calculations, something undesirable in a real-time procedure. To avoid this, the dynamics of the various error states have been chosen as: constants, random walks, or white noise, so the transition matrix is diagonal, and all diagonal elements have values that are either 0 or 1. Multiplying by this matrix means either doing nothing or zeroing some state vector components and the corresponding columns and rows of the covariance matrix.

(II) Stochastic update: the covariance matrix of the system noise is added to the state covariance matrix after step (I). In the navigation filter, as implemented in the software, the system noise covariance matrix is diagonal, except for several small 6x6 blocks along the main diagonal, one for each satellite present in the solution.

(III) Data update: All data not rejected during pre-processing are assimilated in the solution. This step requires computing the coefficients of the error states, or "partials", to form the observation equations matrix.

Because the models of the error state dynamics have been chosen so as to minimize steps (I) and (II), the calculation of those coefficients makes this final step the most time-consuming of all the three update stages.

The filter updates made during the differential real-time solution discussed in the next section, with three baselines (from the rover to three fixed receivers), estimating all Lc biases, four refraction zenith delay corrections, and 102 satellite estates, plus the three instantaneous coordinates of the rover, was about 0.12 seconds, using the 1.7 MHz laptop computer described in the following section. Those filter updates were made once per minute. The instantaneous position solution at every epoch, a considerably smaller calculation, took about 0.02 seconds, and both took much less than the 1 second updates of the incoming data.

Mathematical description:

(a) Hill's perturbation differential equations. The presence of small errors in the known initial position and velocity of a satellite, and of small accelerations not taken into account, or modeled improperly, when calculating its orbit, or ephemerides, will result in an erroneous computed trajectory. The differences in speed and velocity between the true and the computed orbit generally vary quite gradually and tend to be periodic, with a fundamental period equal to the orbit period. For nearly circular orbits with semi-major axes of some 4 terrestrial radii, such as those of GPS and other GNSS satellites, the orbit errors can be described with relatively simple analytical expressions. Probably the simplest are solutions of Hill's differential equations [1, *ib.*]

These equations assume that the only force acting on the satellite is that of the Earth's gravity field, that this field is that of a point mass at the Earth's center, which is also the origin of coordinates, about which the satellite follows a circular orbit at constant speed; that a system of coordinates is used where the x axis always points to the satellite, the y axis is parallel to the velocity vector, and therefore perpendicular to x, and the z axis is perpendicular to the plane of the orbit (defined by the satellite position and velocity vectors or, equivalently, by the x, y axes), and forms a right-handed triad with the other two. So this is a rotating frame of Cartesian coordinates that turns within the orbit plane with the same angular frequency ω as the satellite orbit. This orbit angular frequency is, by Kepler's 3rd Law:

$$\omega = (GM_E/a_s^{3/2}) \qquad (1)$$

Here G = universal gravitational constant, M_E = Earth's mass, a_s = (mean) semi-major axis of the satellite orbit.

Hill's equations are obtained by linearizing Newton's Second Law of motion: *acceleration* = *force/mass*, because even with the much simplified gravity field, the acceleration of the satellite is a non-linear function of position (inverse of distance squared). Linearized differential equations are often used to describe small perturbations in the solutions of nonlinear equations. Here those small perturbations (small compared to the radius and velocity of the orbit), are the orbit errors caused by unavoidable small errors in the initial conditions and in the force models used to calculate the ephemerides. Hill used his equations for calculating the orbit of the Moon. These days they are used to plan spacecraft rendezvous maneuvers and satellite station keeping in "formation flying". Also for precise orbit determination [18], and GNSS navigation [19], [20].To introduce the equations, let x, y, z be the orbital perturbations in the radial, along, and across directions. These are Hill's differential equations:

$$x''(t) - 2\omega y'(t) - 3\omega^2 x(t) = f_x \qquad (2a)$$

$$y''(t) + 2\omega x'(t) = f_y \qquad (2b)$$

$$z''(t) + \omega^2 z(t) = f_z \qquad (2c)$$

where f_x, f_y, f_z are perturbing accelerations along the x, y, z axes, and b'(t) and b"(t) are the first and second time derivatives of function b(t).

(b) The analytical solution. Hilll's equations have the following unforced solution, where x_o, y_o, z_o, x'_o, and y'_o, z'_o are the components of initial position and velocity:

$$x(t) = x'_o \sin(\omega t)/\omega - (3x_o+2y'_o/\omega) \cos(\omega t) + (4x_o+2y'_o/\omega) \qquad (3a)$$

$$y(t) = (6x_o+4y'_o/\omega) \sin(\omega t) + 2x'_o/\omega \cos(\omega t) – (6\omega x_o+3y'o)t + (y_o-2x'_o/\omega) \qquad (3b)$$

$$z(t) = z_o \cos(\omega t) + z'_o \sin(\omega t)/\omega \qquad (3c)$$

$$x'(t) = x'_o \cos(\omega t) + (3\omega x_o+2y'_o) \sin(\omega t) \qquad (3d)$$

$$y'(t) = (6\omega x_o+4y'_o) \cos(\omega t) – 2x'_o \sin(\omega t) - (6\omega x_o+3y'_o) \qquad (3e)$$

$$z'(t)= -z_o\omega \sin(\omega t) + z'_o \cos(\omega t) \qquad (3f)$$

(c) Observation equation coefficients of the satellite unknown error states. Let $\mathbf{r_s}$ be the unit 3-vector pointing from a station to a satellite, and $\mathbf{u_x}$, $\mathbf{u_y}$, $\mathbf{u_z}$ be the *row* unit 3-vectors pointing in the radial, along, and across direction, respectively. Let **J** be the 3x3 matrix with rows equal to $\mathbf{u_x}$, $\mathbf{u_y}$, $\mathbf{u_z}$. respectively. Define a 3-vector $\mathbf{s} = \mathbf{J}\mathbf{r_s}$. The contribution of the orbit errors (perturbations) to the *a priori* range error dr is the scalar product: $dr = \mathbf{s}^T d\mathbf{r}$, where $\mathbf{dr} = [x, y, z]^T$ is the 3-vector orbit position error. This range error dr is a linear combination of x, y, z, themselves linear combinations of x_o, y_o, z_o, x'_o, y'_o, z'_o, according to equations (3a-f). Let the 6-vector **hs** be the vector of the coefficients of the unknown initial satellite error states – *ordered as follows*: z_o, y_o, x_o, z'_o, y'_o, x'_o -- in the observation equations of un-differenced range measurements (i.e. pseudo-range and phase-range) from a station to a satellite. These coefficients can be combined with those for other station and satellite pairs to form the observation equations of single and double differences. To find the observation equation coefficients for the still unknown satellite state, first define:

$$a = \cos(\omega t) \quad (4a) \qquad b = 6(\sin(\omega t)-\omega t) \quad (4b)$$

$$c = 4\cos(\omega t) \quad (4c) \qquad d = \sin(\omega t)/\omega \quad (4d)$$

$$e = 4\omega \sin(\omega t)-3t \quad (4e) \qquad f = 2/\omega\,(1-\cos(\omega t)) \quad (4f)$$

$$g = –f \quad (4g) \qquad h = d \quad (4h)$$

Finally, using equations (4a-h), one arrives to the following observation equation coefficients for the six error state components of the satellite:

$$hs(1) = s(1)a \qquad (5a)$$

$$hs(2) = s(2) \qquad (5b)$$

$$hs(3) = s(2)b + s(3)c \qquad (5c)$$

$$hs(4) = s(1)d \qquad (5d)$$

$$hs(5) = s(2)e + s(3)f \qquad (5e)$$

$$hs(6) = s(2)g + s(3)d \qquad (5f)$$

Where hs(i) and s(i) are the i^{th} components of vectors **hs** and **s**, respectively.

(d) The homogeneous and forced solutions and the pseudo-epoch state. Equations (3a-f) can be written in matrix form as:

$$\mathbf{x}(t) = \mathbf{F}(t-t_o)\, \mathbf{x}_o \qquad (6)$$

where $\mathbf{x}(t)$ and $\mathbf{x}_o$ are the satellite 6-vector state at times t ad t_o, respectively, and **F** is the 6x6 **state transition matrix** between times t_o and t. The elements of this matrix are the coefficients of the elements of $\mathbf{x}_o$ in equations (3a-f). A very useful property of **F,** in systems of linear differential equations with constant coefficients, such as Hill's, can be written as follows:

$$\mathbf{F}^{-1}(t_i-t_o)\, \mathbf{F}(t_i-t_{i-1}) = \mathbf{F}^{-1}(t_{i-1}-t_o) \qquad (7).$$

If the forcing accelerations f_x, f_y, f_z in (2a-c) are all zero, then the homogenous, or unforced response is given by (6), which describes the orbit errors caused only by errors

in the initial states. But, in general, the acceleration errors are not zero. They are caused by both errors in the modeled forces, and by un-modeled forces. For GPS satellites they are quite small, of the order of 10^{-8} m/s^2.
If **f** is the 3-vector with components f_x, f_y, f_z, then the complete, forced solution of Hill's equations is:

$$\mathbf{x}(t) = \mathbf{F}(t-t_o)\,\mathbf{x}_o + \int_{t_o,t_i} \mathbf{F}(t-\tau)\,\mathbf{f}(\tau)\,d\tau \qquad (8).$$

$\int_{t_o,t_i}$ indicates integration between t_o and t.
Let t_{i-1} and t_i be two consecutive epochs in which the filter is updated. The time-invariance properties of Hill's equations make it possible to replace t_o with t_{i-1} in (8) and obtain:

$$\mathbf{x}(t_i) = \mathbf{F}(t_i-t_{i-1})\,\mathbf{x}(t_{i-1}) + \int_{t_{i-1},t_i} \mathbf{F}(t_i-\tau)\,\mathbf{f}(\tau)\,d\tau \qquad (9).$$

If the **pseudo-epoch state** is defined as:

$$\mathbf{z}(t_i) = \mathbf{F}^{-1}(t_i-t_o)\,\mathbf{x}(t_i) \qquad (10),$$

then

$$\mathbf{z}(t_i) = \mathbf{F}^{-1}(t_i-t_o)\,[\mathbf{F}(t_i-t_{i-1})\,\mathbf{x}(t_{i-1}) + \int_{t_{i-1},t_i} \mathbf{F}(t_i-\tau)\,\mathbf{f}(\tau)\,d\tau]\,.$$

But, according to equation (7), this also can be written as

$$\mathbf{z}(t_i) = \mathbf{F}^{-1}(t_{i-1}-t_o)\,\mathbf{x}(t_{i-1}) + \mathbf{G}(t_i) \qquad (11),$$

where

$$\mathbf{G}(t_i) = \mathbf{F}^{-1}(t_i-t_o) \int_{t_{i-1},t_i} \mathbf{F}(t_i-\tau)\,\mathbf{f}(\tau)\,d\tau\,. \qquad (12).$$

But the first term in (11) is $\mathbf{z}(t_{i-1})$. So, finally:

$$\mathbf{z}(t_i) = \mathbf{z}(t_{i-1}) + \mathbf{G}(t_i) \qquad (13)\,.$$

Equation (13) defines, in matrix form, six scalar *difference* equations that express the dynamics of the orbit error. This is the appropriate way to formulate system dynamics when working with discrete-time estimators such as the navigation Kalman filter.
When $\mathbf{f}(t) = \mathbf{0}$ (a null 3-vector) for all t, then from (10) it follows that $\mathbf{x}(t_i) = \mathbf{F}(t-t_o)\,\mathbf{z}(t_i)$, so $\mathbf{z}(t_i) = \mathbf{x}_o$ and the pseudo-epoch state becomes identical with the initial state, which is a constant vector. In general, $\mathbf{f}(t)$ is not zero, and the pseudo-epoch state changes value from one epoch to the next, unlike a true initial epoch state.
With this formulation, the system's transition matrix in (13) is the identity matrix. This means that no arithmetic operations are required for the deterministic update of the satellite states partition of the full state space vector (that contains all of the solution unknowns: orbit parameters, residual zenith delays, perhaps some site coordinates, and the position of the rover), or for the deterministic update of the corresponding partition of the filter covariance matrix. Instead, most of the update-related arithmetic is dedicated to computing the coefficients in the observation equations of the satellite states according to (4a-h) and (5a-f). This can be done quickly, because the expressions involved are sums of low degree polynomials, sines and cosines, for which compilers have very efficient in-line subroutines.

(e) The stochastic model. The mathematical model for the stochastic update of the filter is based on two ideas:
(1) the forcing accelerations in (2a-c) change very slowly.
(2) It is most desirable *not* to have to add new error states just to represent those accelerations. The total number of satellite states, which equals the number per satellite times the number of satellites in the solution, must be kept to a minimum, to reduce computing overhead – so as not to slow down real-time processing.
To satisfy (2) above, the author has made the following compromise between realism and computing speed:
(a) The three forcing functions, and therefore the vector **f**, are assumed to stay constant over several filter updates, and then switch at the same time to new values.
(b) The amplitudes of the piece-wise constant forcing functions are random numbers with *a priori* standard deviations of 10^{-8} m/s^2.
(c) The pseudo-epoch states of all satellites have their stochastic updates together, and only when the forcing functions switch to new values. By trial and error, it has been found that 20 minutes is a good interval between switches.
Because of these approximations, the number of error states per satellite does not have to be increased, and remains at six.
Let:

$$\mathbf{Q_s} = \mathbf{G}\,\mathbf{Q_f}\,\mathbf{G}^T \qquad (14)$$

Here $\mathbf{Q_s}$ is a 6x6 sub-matrix representing the increase in uncertainty in the pseudo-epoch states of a satellite at each of their successive updates. **G** is no longer as given by (12); instead it is:

$$\mathbf{G}(t_i) = \mathbf{F}^{-1}(t_i-t_o) \int_{t_{iPrevious},t_i} \mathbf{F}(t_i-\tau)\,\mathbf{f}_c\,d\tau \qquad .(15)$$

Now $\mathbf{f}_c$ is a 3-vector with components a_x, a_y, a_z , which are the constant amplitudes of the accelerations f_x, f_y, f_z in the interval $t_{Previous} < t \leq. t_t$ The integration limits in the second term are: $t_{Previous}$, the last time the components of $\mathbf{f}_c$ changed values (several filter updates ago), and the present update epoch t_i, with $t_i - t_{Previous} \sim 20$ minutes. The elements of the integral of **F** in (15) are the integrals of the time-dependent coefficients of (3a-f), *so they are also combinations of sines, cosines, and low-degree* polynomials, and can be calculated quickly.

Finally, $\mathbf{Q_f}$ is the diagonal 3x3 covariance matrix:

$$\mathbf{Q_f} = E\{\mathbf{f}\,\mathbf{f}^T\} \qquad (16)\,.$$

Summing up: according to equation (13) the deterministic filter update is a “do nothing” step (multiplying the pseudo-epoch state by a unit matrix). Equations (14), (15) and (16) are used in the stochastic update of the 6 pseudo-epoch states of each satellite, to compute the elements of the 6x6 matrix $\mathbf{Q}_s$ to be added to the covariance matrix of the full state vector. This operation has to be repeated as many times as there are satellites in the solution. Finally, equations (3a-f), (4a-h), and (5a-f) are used to calculate the observation equation coefficients (or “partials”) of the states of each satellite.

Broadcast ephemerides updates. The parameters of the orbits in the GPS Navigation Message are updated every two hours on the even hour (regular updates), and occasionally a minute before the hour (unscheduled updates). When estimating orbit errors, it is not absolutely necessary to start using the new orbit parameters every time they are updated, unless they are very different from the old ones. In that case, the orbit error states must be relaxed in the filter during the stochastic step by adding system noise in proportion to how much the orbit changes when calculated with the new parameters.

Initial satellite state uncertainties. The initial, or *a priori*, uncertainties of the pseudo-epoch states (one standard deviation), are the same used in solutions made with the post-processing software: position, 4 m, velocity, 6 mm/s -- these values are per coordinate -- and their covariance matrix is diagonal.

THE IBERIAN REAL-TIME TEST

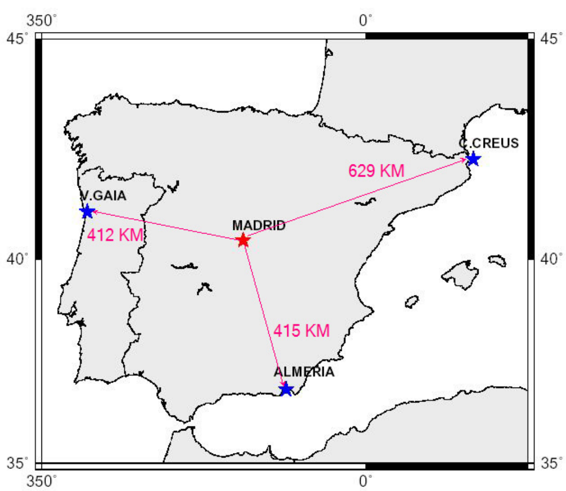

Figure 3. Location of the four sites used in the experiment, and length of the baselines from the three EUREF “reference stations” to the “rover”, the ITACyl site in Madrid.

The Data. In 28 April 2008, the author conducted a test at his home in College Park, Maryland, USA, receiving and processing in real-time NTRIP data streamed from four sites in the Iberian Peninsula: one in Portugal (GAIA), near the city of Porto, and another three in Spain (CREU, ALME, and MDRD) located in Cape de Creus, and in the cities of Almeria and Madrid, as shown in Figure 3.
This test lasted five hours; all the data were collected, transmitted, received, and processed at 1 Hz.
Note. In actual applications, the data analysis could be split, as in the Virtual Reference Stations (VRS) approach, between the network operators and the users, with the former calculating correctors with data from their network and transmitting them to the users, so these can correct their roving receiver data and calculate the precise position of their vehicles. But for this test, the data from all receivers has been processed in a single, unified solution.
The coordinates of the ITACyl site in Madrid were not given in the same terrestrial reference frame as the three EUREF sites, so it was positioned relative to them in a static differential solution, using observations from a different day than that of the test. The solution was made with the author’s precise post-processing software “IT”; the precision per coordinate should be better than 2 cm.
The four sites were equipped with the following receiver/antenna combinations (each listed next to the corresponding station Network IDs):

GAIA: Leica RS500/LEIAT504,
ALME, CREU: Trimble NetrS/TRM41249.00,
MDRD: Trimble Netr5/TRM29659.00.

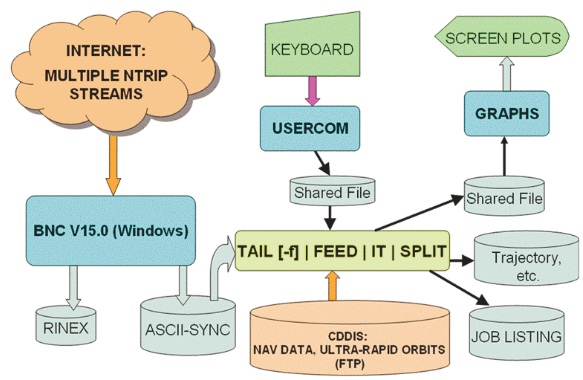

Figure 4. Data processing flow diagram.

A Real-Time Data Processing System.
Figure 4 shows the flow diagram of the data processing for this experiment, as implemented by the author, at home, in an Intel Centrino 1.7 Mhz laptop, with 1 Gb of RAM, running Windows XP. All programs, except for BNC and tail are written in Fortran, compiled with Compaq Visual Fortran (VF) Version 6.6 -- a Fortran 95 compiler. The data are received from the Internet using

the NTRIP Client **BNC Version 1.5,** for Windows, distributed by BKG. This program collects all the data obtained at any given epoch at those stations whose streams it is receiving from an NTRIP caster. It waits for data for up to one second before saving those that have arrived to an ASCII file, called “ascii.sync” in the diagram. (It also saves data in RINEX files, one for each station, but those were not used here.) The received data accumulates, one epoch at the time, in “ascii.sync”. The Microsoft Windows version of the Unix program “**tail**” reads in non-interactive mode (with switch “-f”) the latest data records added to this file, and pipes them to the standard input of program “**feed**”. This program also receives control messages from the user through “**usercom**” and passes them to the rest of the system, providing an interactive interface. Program “**feed**” then converts the GPS measurements saved in “ascii.sync” in a format closely related to RTCM, to the RINEX format used in the navigation software. It also collects, from the geodetic data base of the CDDIS at NASA Goddard, the SP3-formmated file with the latest predicted ultra-rapid orbits from the IGS, and the latest “hourly” RINEX navigation file. Then “feed” passes all this information, some directly through a pipe, and some through shared files to the navigation software (real-time) “**IT**”. There the data first are preprocessed, and then are analyzed to produce the precise real-time kinematic solution. “IT” also sends information on the results obtained so far to program “**graphs**”, through a VF “SHARED” file, to be plotted on the computer monitor, so the user can have some idea of how things are going. The output of “IT” contains records with the latest estimated position of the rover both in x, y, z EFEC coordinates and Up, East, North topocentric coordinates, and their corresponding 9x9 covariance matrices. It also contains the records of the run’s listing: the full history of what happened during the data processing. Program “**split**” receives the standard output of “IT”, separates the run listing from the results, and saves them in two separate output files.

Results. The test real-time calculations were made with GPS orbits taken from the IGS precise predicted ultra-rapid orbits, downloaded from the CDDIS during the run. Only GPS observations were used. Saved in the file “ascii.sync”, they were used again later, in two off-line runs with the same real-time software, but now using the GPS Navigation Message broadcast ephemerides instead of the IGS ultra-rapid orbits, first unchanged, and then corrected with the procedure described previously.

Figure 5 shows the comparison of the instantaneous kinematic position of MDRD compared to its known position. 5(a) is the result with the broadcast ephemerides kept fixed in the navigation solution; 5(b) is the solution with the broadcast ephemerides corrected (adjusted) in the solution; 5(c) is the real-time result, with the predicted ultra-rapid IGS orbits, kept fixed in that solution.

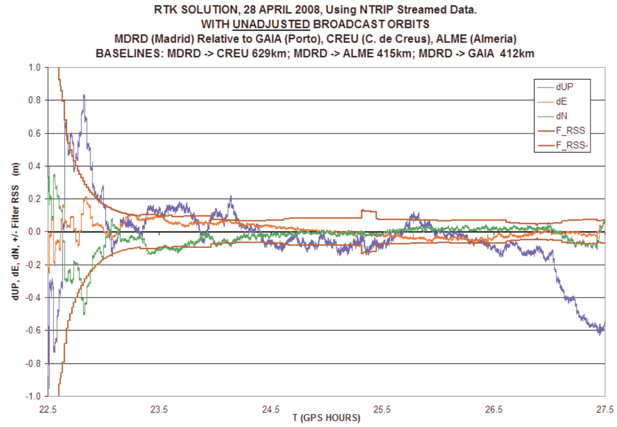

Figure 5(a) Comparison between the known, or “truth”, position of MDRD and the instantaneous position estimated with the uncorrected Broadcast Ephemerides. The brown symmetric lines show the *a posteriori* (± RSS) or three-dimensional formal precision of the estimated position at each epoch.

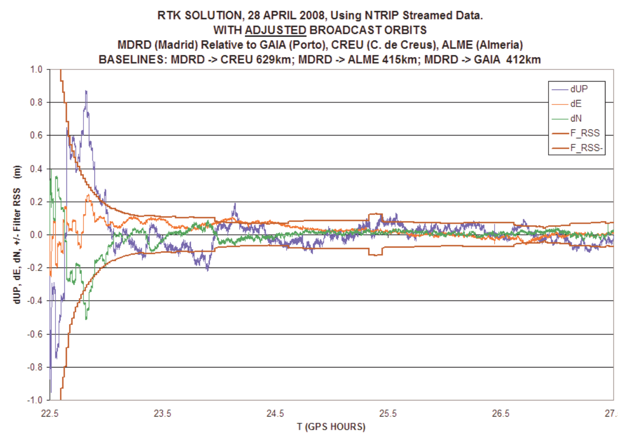

Figure 5(b) As in (a), but with the errors in the Broadcast Ephemerides corrected in the solution.

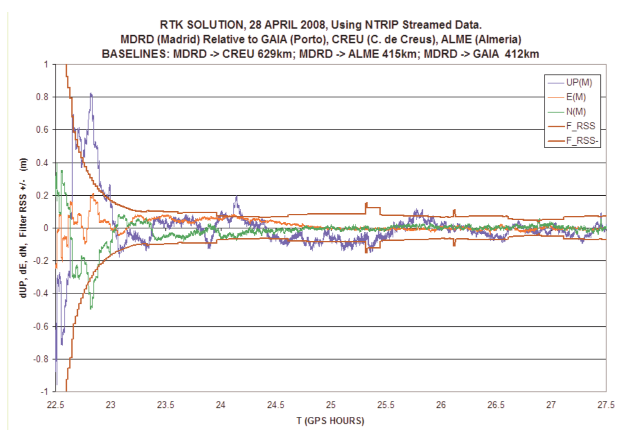

Figure 5(c) As in (a), but using the more precise ultra-rapid predicted IGS orbits (the actual real-time results).

The first plot shows a rather large excursion of more than half a meter in height, away from the precise coordinates of MDRD -- the test's "truth" -- when using the uncorrected broadcast ephemerides. This excursion is also in clear disagreement with the size of the "error bars" of those position estimates (the two brown lines). The second plot shows that a considerable improvement is gained by correcting the broadcast orbits in the solution. The third plot shows that using the IGS precise predicted orbits gives slightly better results than adjusting the broadcast orbits. In both Figs. 5(b) and 5(c) the discrepancies between the kinematic position and "truth" are never larger than twice the formal 1-STD (RSS) error bars, and stay mostly within them.

The Filter Convergence Problem. Figure 6 shows the result of analyzing the same data with the same ultra-rapid orbits and navigation message file as in the real-time solution (Fig. 5(c), above). The difference here is that this solution was made off-line: it is a post-processed solution, using first the navigation filter, as in real time, and then a smoother. In this type of solution, all the data collected during a session is implicitly used to estimate all of the unknowns. So the precision of the estimated rover position tends to be equally good at all epochs (the smoother algorithm "smoothes out" the precision as a function of time).

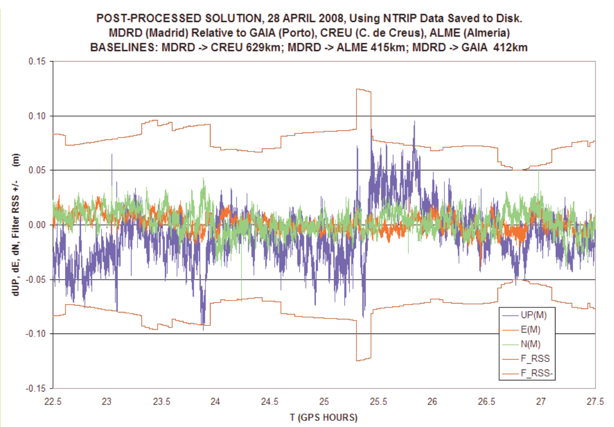

Figure 6. Same as Fig. 5(c), but for a post-processed, kinematic navigation solution obtained analyzing the data with a Kalman filter and a smoother.

The values of nuisance unknowns present in many consecutive observations, such as the Lc biases in the carrier phase observations of a given satellite, are estimated much better with the filter towards the end of a period in which that satellite is present. Consequently, the precision of the whole solution becomes better towards then end of that period. At the beginning of the real-time filter run, all error states are poorly known, since such knowledge is limited to whatever preliminary information might be available about them, usually with large *a priori* uncertainties. As the run progresses, the assimilation of ever increasing amounts of data, epoch after epoch, contributes more and more information, reducing the uncertainties, including those on the instantaneous position of the rover, which improves as time goes by.

This is clearly shown both by the size of the discrepancies between kinematic and "truth" position, and by the brown RSS lines in the plots. The initial period of large discrepancies and of large formal RSS uncertainties in the rover's instantaneous position, is the *filter convergence period*. It is characteristic of filter-only, wide-area, point-positioning or long-baseline differential solutions, including all such real-time kinematic solutions, where it can be a limiting problem if precise results are needed soon after the start of a session.

Had one already known, at the beginning of the real-time session, the values of the nuisance unknowns (Lc biases, etc.) as well as one will know them at the end, after the filter has assimilated all the data available, then the initial results might have been as good as the final ones.

There are several ways of shortening the convergence period in real-time, wide-area solutions, but all of them require additional information, often of high-quality, and not always available, besides the receiver data.

For example:

(a) Resolving the carrier-phase ambiguities to remove the Lc biases from the ionosphere-free carrier phase linear combination, which requires making very precise ionospheric corrections to the data [1, *ib.*]

(b) Starting at a place of well-known coordinates, and conditioning the solution with those coordinates [5 *ib.*]

(c) If the GNSS receiver is on a buoy, ship, or some other floating platform, the solution can be conditioned by taking advantage of the fact that, after filtering out the vertical motion caused by ordinary waves, remaining changes in sea surface (and receiver antenna) height are due to tides, air pressure changes, wind pile ups, drifting eddies, etc., and tend to be quite gradual [6, *ib.*]

Real Time or Post-Processing? When deciding how to analyze their GNSS data, users must consider both real-time processing and post-processing, and choose one of them according to what they want to do. If the precise results are needed to steer a vehicle, as in precision farming, then real-time processing is the only choice. However, if what is needed is the very best estimated position of the rover that can be obtained from the data, and there is no pressing need to get the results during the survey, then post-processing is the better option. As shown in Fig. 5(c), if the session is long enough to acquire sufficient data, post-processing is likely to provide uniformly good results without all the additional information needed to speed up the convergence of the filter in real time. It is also a more robust approach, because the data can be processed more than once, in somewhat different ways, to find out one that gives the

best results. It is also more forgiving of operator errors, while in real-time solutions everything has to be done correctly right away, every time, all the time. While much progress has been made over the years in facilitating the use and ensuring the success of precise real-time navigation solutions, post-processing is by nature more precise and reliable.
Real-time is for users that cannot work in some other way, or are sure that getting their results very quickly is worth taking a somewhat greater risk.

CONCLUSIONS

The rapid increase in recent years in the number of real-time GNSS Internet data streams available from many sites and often free of charge to all those interested in using them, is an unprecedented phenomenon that should have far-reaching repercussions on the practice of satellite positioning. The free GNSS data streams, in particular, give developers and inventors access to a test-bed that is as large as the whole world. Such ready availability of real-time data from so many different places, with very different receiver environments, is spurring the fast development of navigation and surveying techniques, the introduction of new applications of satellite-based positioning, and helping more people take advantage of them.

Central to these remarkable developments is the adoption of a set of common standards for the collection, dissemination, and use of the streamed data, of which the NTRIP protocol is a very important component.

As shown in this paper, it is possible, with the support of a properly designed wide-area GNSS network of reference sites, and with an appropriate navigation procedure, to achieve decimeter-level precision in real-time navigation, even hundreds of kilometers away from any network station.
There are precise orbits, such as the IGS predicted ultra-rapid orbits, that can be downloaded for free over the Internet, with anonymous FTP, from sites such as NASA's CDDIS, for use in differential real-time positioning.
It seems also likely that in the near future there will be real-time satellite clock corrections available as well, for precise real-time point positioning over large areas [21].

It is possible to get precise differential solutions with just the broadcast ephemerides, by using the procedure described in this paper. This means that differential navigation with the support of a wide-area network of land receivers can be quite self-contained, since it is not essential to have access to very precise orbits. So the technique described here may provide an alternative way of doing things, or at least a fall-back option for those times when the precise orbits are not available.

Finally, a basic problem with real-time positioning is how long it takes to collect the data needed to achieve precise results, particularly in kinematic mode. There are ways to mitigate this problem (e.g.: resolving phase ambiguities with very good ionospheric corrections), but they require additional information that sometimes can be hard to get, so finding new practical remedies to the convergence problem remains an important challenge.

ACKNOWLEDGEMENTS

The real-time navigation software "IT" used to analyze the GPS data for the Iberian test has been developed as part of a precise real-time positioning project directed by Dr. Alan G. Evans at the NSWC, Dahlgren Division.
The author thanks him for his leading role in making the project happen and keeping it going, and also thanks the NSWC staff members Andrew Sutter -- who is in charge of other components of the real-time system -- for carrying out a number of multi-day tests to debug the real-time navigation software, and James P. Cunningham, also involved in this project, for practical advice and incisive comments.
The author thanks Georg Weber, from BKG, in Frankfurt, for providing the BNC NTRIP client software, and explaining its use, and Alberto Garcia-Rigo from UPC, in Barcelona, for helping him subscribe to the data streams for the Iberian test. And many thanks to those responsible for the EUREF and Castilla-Leon networks, and NOAA's NGS test network, for providing free access to their NTRIP data streams, and making possible this study.

REFERENCES

[1] Colombo, O. L., Hernandez-Pajares, J. M. Juan, J. Sanz, "Wide-Area, Carrier-Phase Ambiguity Resolution Using a Tomographic Model of The Ionosphere", *Navigation*, Vol. 49,No. 1, 61-69,Spring 2002 Issue, December 2002.
[2] Colombo, O.L., "Errors in Long Distance Kinematic GPS", *Proceedings of the ION GPS '91*, Albuquerque, N.M., September 1991.
[3] Colombo, O. L., A. G. Evans, "Testing Decimeter-Level Differential Kinematic GPS Over Great Distances at Sea and on Land", *Proceedings of the ION GPS '98*, Nashville, Tennessee, September1998.
[4] Hill, G. W., *Am. J. Math.* Vol. **1,** 5, 129-245, 1878.
[5] Colombo, O.L., A.W. Sutter, A.G. Evans, "Evaluation of Real-Time, Long-Range, Precise, Differential, Kinematic GPS Using Broadcast Orbits", *Proceedings ION GNSS-2003*, Portland, Oregon, September 2003.

[6] Colombo, O.L., Evans, A.G., Ando, M., Tadokoro, K., Sato, K., T. Yamada, "Speeding Up The Estimation Of Floated Ambiguities for Sub-Decimeter Kinematic Positioning at Sea", *Proceedings ION GPS 2001*, Salt Lake City, September 2001.

[7] Gebhard, H., Weber G.: "Networked Transport of RTCM via Internet Protocol", Design-Protocol-Software, published by Federal Agency for Cartography and Geodesy (BKG), June 2003

[8] Peterzon, M., "Distribution of GPS-data via Internet", Thesis work report, National Land Survey of Sweden, Lantmäteriet, Gävle, Sweden, 2004.

[9] González-Matesanz, F.J., Weber, G., Celada, J. Dalda, A. y Quiros, R. "EUREF-IP Project. Results with GPRS", *Proc. 4ª Assembleia Luso Espanhola de Geodesia e Geofísica*, Figueira da Foz, Portugal, 2004

[10] Henning, W., "Real Time Positioning and the Role of the NGS", Presented at the CORS Users Forum, CGSIC-47th Meeting, Fort Worth, September 2007.

[11] "Ntrip Documentation", "Ntrip Implementation", both PDF files, available from BKG at: http://igs.bkg.bund.de/index_ntrip.htm

[12] Colombo, O.L., "Long-Range Kinematic GPS", In *GPS for Geodesy*, P. Teunissen and A. Kleusberg, Editors, Springer Verlag, Lecture Notes in Earth Sciences, 1998.

[13] Zumberge, J.F., Heflin, M.B., Jefferson, D.C., Watkins, M.M., F.H. Webb, "Precise Point Positioning for the Efficient and Robust Analysis of GPS Data from Large Networks", *J. Geophys. Res.*, 102, 5005-5017, 1997.

[14] Zumberge, F.M., M. M. Watkins, F. H. Webb, "Characteristics and Applications of Precise GPS Clock Solutions Every 30 Seconds", *Navigation*, Vol.44, No. 4, pp. 449-456, Winter Issue 1998-1999, December 1998.

[15] Kouba, J., P. Heroux, "Precise Point Positioning Using IGS Orbit Products", *GPS Solutions*, Vol. 5, No. 2, pp12-28, Fall of 2000.

[16] Yunck, T. P.,W. I. Bertiger, S. C.Wu, Y. E. Bar-Sever, E. J. Christensen, B. J. Haines, S. M. Lichten, R. Muellerschoen, J, Vigue, Y, P.Willis. 1994. "First assessment of GPS-based reduced dynamic orbit determination on TOPEX/Poseidon." *Geophys. Res. Lett.* 21(7):541–544, 1994.

[17] G. J. "Biermann. Factorization Methods for Discrete Sequential Estimation". Academic Press, 1977.

[18] Colombo, O.L., S.B. Luthcke, "Kinematic Point Positioning of a LEO with Simultaneous Reduced-Dynamic Orbit Estimation", *Proceedings ION GNSS-2004*, Long Beach, California, September 2004.

[19] Colombo, O.L., "The dynamics of GPS Orbits and the Determination of Precise Ephemerides, *J. Geophys. Res.*, 94, B7, 9167-9182, 1989.

[20] Colombo, O.L., "Ephemeris errors of GPS satellites", *Bull. Geodesique*, Vol. 60, No. 1, Paris, 1986.

[21] Weber, G., Mervart, L., Lukes, Z., C. Rocken, "Real-time Clock and Orbit Corrections for Improved Point Positioning via NTRIP", *Proceedings of the ION GNSS 2007*, Fort Worth, September 2007.